



10P+CJC8910 适配问题及解决方法

一、问题描述

1.1 CJC8910 为什么不能直接替代 8311

CJC8910 与 8311 脚位是共用的，但寄存器不同，所以需要主控 IC 调试软件来达成。

1.2 调试过程中出现的问题及当前调试结果

问题一：

客户使用嵌入式 Linux 系统读不出寄存器地址和寄存器值，原因有三：

- ① 读取寄存器过程中，需要保持 MCLK 持续存在状态
- ② 读取的寄存器地址错误
- ③ 嵌入式 Linux 下 i2c-tool 工具的使用方法

解决问题 ① 读取寄存器过程中，需要保持 MCLK 持续存在状态

CJC8910 的 I2C 配置接口为同步 I2C，请在系统上电的第一时间给出主时钟 MCLK，然后再进行 I2C 配置。

调试过程：即在配置寄存器时,必须先一直给主时钟 MCLK:12.288MHz。

解决问题 ② 读取的寄存器地址错误

写的地址要正确，CJC8910 使用 2 线 I2C 格式配置寄存器如下表所示，最后根据客户芯片CE脚的状态为1和使用的Linux系统读取地址为7 位设备地址 (十六进制)得出其地址为：0X19

Item	芯片CE脚状态	8 位设备地址写 (十六进制)	7 位设备地址 (十六进制)
Value	0	0x30	0x18
Value	1	0x32	0x19

解决问题 ③ 嵌入式 Linux 下 i2c-tool 工具的使用方法

直接使用 i2cdump 命令

i2cdump：读取指定设备的全部寄存器的值。

```
i2cdump [-f] [-r first-last] [-y] i2cbus chip-address [mode [bank [bankreg]]]
```



参数说明:

参数 r: 指定寄存器范围, 只扫描从 first 到 last 区域;

参数 f: 强制访问设备。

参数 y: 关闭人机交互模式;

参数 i2cbus: 指定 i2c 总线的编号

参数 address: 指定设备的地址

参数 mode: 指定读取的大小, 可以是 b, w, s 或 i, 分别对应了字节, 字, SMBus 块, I2C 块

实例

```
root@imx6ul7d:~# i2cdump -f -y 1 0x1a
No size specified (using byte-data access)
 0 1 2 3 4 5 6 7 8 9 a b c d e f 0123456789abcdef
00: 89 00 00 00 00 00 00 00 00 96 00 00 00 00 00 00 ?.....?.....
10: 00 00 00 00 8c 0c 00 00 00 00 00 00 00 00 00 00 .....??.....
20: 00 00 00 00 00 00 00 00 01 32 00 00 00 00 00 00 .....?2.....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40: 00 00 00 00 00 0a 08 00 00 00 00 00 00 30 01 00 .....??.....0?.
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
60: 00 00 00 1f 2b 01 09 2b 00 01 1c 00 00 00 00 00 ...?+??+.??.....
70: 00 00 00 00 00 00 00 2e 00 00 00 00 00 00 00 00 .....
80: 03 00 00 00 00 00 00 00 00 00 00 00 0f 04 00 1e ?.....???.?
90: f1 0b 01 1c f3 0a 05 16 f8 07 11 05 05 40 00 00 ??????????????@..
a0: 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 .....?.....
b0: 00 00 00 00 60 48 00 00 00 00 00 00 00 00 00 00 .....`H.....
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
root@imx6ul7d:~# █
```

注: 1 是总线号, 0x19 是设备地址。



问题二:

CJC8910 在读取寄存器过程中为什么与 8311 读取的不相同。

音频芯片寄存器地址一般是 8 位，寄存器值一般也是 8 位，而 CJC8910 寄存器地址一般是 7 位，而寄存器值是 9 位，大家对 8 位寄存器地址+8 位寄存器值读写较熟，但是对于 7 位寄存器地址+9 位寄存器值读写不是很理解，这里通过 I2C 通信读写寄存器举例说明：

①8311 寄存器的配置方法和数据构成：

8 位寄存器地址+8 位数据形成 16 位数据，写的时候都是按字节写的。

Start+设备地址+Write (0) +ack+寄存器地址 (8-bit) +ack+寄存器数据 (8-bit) +ack+stop

驱动中对应 i2c write(寄存器地址 (8bit) , 寄存器数据 (8bit))

举例写 8 位寄存器 8 位寄存器数据，应用程序中应该是这样

R0 = 0x34 i2cwrite(0x0, 0x34) i2cwrite(寄存器地址 (8bit) , 寄存器数据 (8bit))

R1 = 0x46 i2cwrite(0x1, 0x46) i2cwrite(寄存器地址 (8bit) , 寄存器数据 (8bit))

R9 = 0x57 i2cwrite(0x9, 0x57) i2cwrite(寄存器地址 (8bit) , 寄存器数据 (8bit))

举例读 8 位寄存器 8 位寄存器数据，应用程序中应该是这样

R0 = 0x34 i2cread(0x0, ®value) 读出后 regvalue = 0x34

R1 = 0x46 i2cread(0x1, ®value) 读出后 regvalue = 0x46

R9 = 0x57 i2cread(0x9, ®value) 读出后 regvalue = 0x57

②CJC8910 寄存器的配置方法和数据构成：

7 位寄存器地址+9 位数据形成 16 位数据，写的时候都是按字节写的。

那么一个寄存器的配置方式如下：

(7 位地址+数据最高位 1 位形成一个字节) + (数据后 8 位一个字节)

举例写 7 位寄存器 9 位数据地址，应用程序中应该是这样

R0 = 0x134 i2cwrite(0x1, 0x34) i2cwrite(寄存器地址 (7bit) +寄存器数据最高位 (1bit) , 寄存器数据低 (8bit))



R1 = 0x046 i2cwrite(0x2, 0x46) i2cwrite(寄存器地址 (7bit) + 寄存器数据最高位 (1bit) , 寄存器数据低 (8bit))

R9 = 0x157 i2cwrite(0x13, 0x57) i2cwrite(寄存器地址 (7bit) + 寄存器数据最高位 (1bit) , 寄存器数据低 (8bit)) 读 0x0E 的值为 0x4A, 读 0x0F 地址的值为 0x00, 其意思为读到 R7 寄存器的数据为 0X04A

举例读 7 位寄存器 9 位数据地址 , 应用程序中应该是这样

R0 = 0x134 i2cread(0x0, ®value) 读出后 regvalue = 0x34

i2cread(0x1, ®value) 读出后 regvalue = 0x1 读取寄存器需要读取两次

R1 = 0x046 i2cread(0x2, ®value) 读出后 regvalue = 0x46

i2cread(0x3, ®value) 读出后 regvalue = 0x0 读取寄存器需要读取两次

R9 = 0x157 i2cread(0x12, ®value) 读出后 regvalue = 0x57

i2cread(0x13, ®value) 读出后 regvalue = 0x1 读取寄存器需要读取两次

问题三:

给客户调配分频关系的方法

确定以下几个主控配置, 才能确定寄存器 R7 (地址: 0X0E) 这个配置

- ① 主控 10P 是做从机还是做主机
- ② 主控的音频数据传输格式, 比如说 DSP Mode IIS Format Left justified
- ③ 音频数据位数, 比如说 16bits 20bits 24bits 32bits

根据客户给的配置①主控 10P 做从机 ②IIS Format ③16bits

从而推测出 R7 (地址: 0X0E) 应该配置为: 0X42

再确定寄存器 R8 (地址: 0X10) 的配置

- ① 主时钟 mclk
- ② 位时钟 BCLK
- ③ 帧时钟 LRCK (采样率)

根据客户给的配置①mclk: 12.288MHz ②BCLK: 未告知 ③LRCK:8 KHz

由于客户未提供 BCLK 的频率, 推测 R8 (地址: 0X10) 的配置有可能为:

0X46 0X66 0X76 其中一个, 最后客户在试 0X76 之后, 播放声音正常, 录音也正常。